|  |
| --- |
| Laboratorio de Computación  Salas A y B |
|  |

|  |  |
| --- | --- |
| Profesor(a): | García Morales Karina |
| Asignatura: | Fundamentos de Programación |
| Grupo: | 132 |
| No de Práctica(s): | 3 |
| Integrante(s): | Viveros Fernández Salvador Juan |
|  |  |
|  |  |
|  |  |
| No. de lista o brigada: | 54 |
| Semestre: | Primero |
| Fecha de entrega: | 04/09/2024 |
| Observaciones: |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Solución de problemas y Algoritmos.**

**Objetivo:**

El alumno elaborará algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

**Desarrollo:**

**Ciclo de vida del software:**

De acuerdo con la ISO, se describe al ciclo de vida del software como el marco de referencia que contiene las actividades y tareas involucradas en el desarrollo, explotación y mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso. Así, los pasos entro de este ciclo son:

-La definición de la necesidad que se debe cubrir.

-El análisis de los datos de entrada necesarios y los conjuntos de salida deseados.

-El diseño del algoritmo con el que se creará el programa.

-La codificación del programa dicho.

-La comprobación de que funcione correctamente.

-La validación de que se arrojen los resultados deseados.

-El mantenimiento y la actualización del programa para su correcto funcionamiento en el tiempo.

**Solución de problemas:**

En la etapa de análisis se desea entender la necesidad a cubrir, es decir, comprender el problema que se desea resolver.

Llevar a cabo este paso es vital para poder crear un software de calidad, puesto que, si no se conoce cuál es la problemática o situación con la que se está lidiando, resulta imposible dar con alguna solución, no obstante, resulta común caer en alguna ambigüedad producto del mal entendimiento del planteamiento inicial.

Por lo tanto, para llevar a cabo este análisis y posterior solución del problema, se es necesario reconocer dos principales conjuntos de datos dentro del sistema de la necesidad específica:

**Conjuntos de entrada:**

Los datos de entrada son todos aquellos datos que se requieren y, por lo tanto, son de los cuales es sistema se va a alimentar para poder funcionar.

**Conjuntos de salida:**

Es todo aquel resultado que el programa arrojará dados los datos de entrada y todo el proceso algorítmico que se llevará a cabo con ellos, siendo así el resultado.

**Algoritmo:**

Se define a un algoritmo como un conjunto de reglas expresadas en lenguaje para completar una tarea en específico, pudiendo ser aplicados una cantidad ilimitada de veces. Así mismo, tiene 3 módulos básicos, el módulo de entrada, siendo estos los datos; el módulo de procesamiento, siendo las operaciones a llevar a cabo y, el módulo de salida, el cual muestra los resultados obtenidos.

Una vez analizado y entendido el problema, se deben de proponer los pasos viables para, dada la información solicitada, poder dar el resultado deseado, siendo esto la creación del algoritmo. Así, en principio si un problema matemático si es posible resolverlo por medio de una computadora, puede ser computable.

Siendo la parte más importante y durable dentro de las ciencias computacionales, siendo independiente del lenguaje de programación y del equipo físico que lo va a ejecutar. Teniendo como características que: debe de indicar el orden de cada paso sin tener ambigüedades, que, si se sigue dos veces o más, se debe conseguir el mismo conjunto de salida, tener un fin y na cantidad finita de pasos, debe cumplir con el objetivo, tener por lo menos una salida perceptible, debe de ser sencillo y legible, realizar su tarea en el menor tiempo posible y que debe de producir el efecto esperado. Cumpliendo estos puntos, se puede considerar que es un software de calidad.

**Pruebas de escritorio:**

Todo algoritmo requiere el uso de variables en el conjunto de entrada para poder funcionar, ya sean numéricos o no numéricos, además de poder ser usados por almacenar generadores de procesos o datos de salida, así, a partir de estos datos, el algoritmo puede fluir en la secuencia de pasos a seguir.

No obstante, dependiendo del tipo de problema, no todos los valores en una variable funcionan en un algoritmo, adquiriendo el nombre de valores críticos.

Para saber si un algoritmo funciona de manera correcta, con los valores estipulados, mientras que no de una salida con valores críticos, se utilizan las pruebas de escritorio, las cuales constan de una matriz donde se especifica el número de iteración o intento, los valores de entrada y la salida que arroja el programa, des dar los resultados deseados, se considera que el algoritmo es correcto.

**Ejercicios:**

**Problema 1: Determinar si un número es positivo o negativo**

Datos de entrada: Número real.

Datos de salida: Si el número es positivo o negativo.

Restricciones: El menor debe ser diferente a 0

Algoritmo:

1-Inicio

2-Solicitar el número

3-Si el número es igual a cero, volver a paso 2, de no serlo, ir a paso 4.

4-Si el número es mayor a 0, ir a paso 6, de no serlo, ir a paso 5

5-Si el número es menor a 0, ir a paso 7

6-Se afirma que el número es positivo, ir a paso 8

7-Se afirma que el número es negativo, ir a paso 8

8-Fin

|  |  |  |
| --- | --- | --- |
| Iteración | Dato de entrada | Salida |
| 1 | 0 | - |
| 2 | 1 | Es negativo |
| 3 | -4 | Es positivo |

**Problema 2: Obtener el mayor de dos números reales**

Datos de entrada: número real 1, número real 2

Datos de salida: Indicación de cual número es mayor

Restricciones: Los números deben ser diferentes de sí mismos

Algoritmo:

1-Inicio

2-solicitar número real 1, ir a paso 3

3-Solicitar el número real 2, ir a paso 4

4-Validar los números, si son iguales, ir a paso 2, de no serlo, ir a paso 5

5-Si el resultado de la resta del primer número, menos el segundo número es positivo, ir a paso 7, de no serlo, ir a paso 6

6-Si el resultado de la resta del segundo número menos el primer número es positivo, ir a paso 8

7-Se afirma que el primer número es el mayor, ir a paso 9

8-Se afirma que el segundo número es mayor, ir a paso 9.

9-Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Número de entrada 1 | Número de entrada 2 | Salida |
| 1 | 5 | 5 | - |
| 2 | 7 | 5 | El primer número es mayor |
| 3 | 8 | 10 | El segundo número es mayor |

**Problema 3: Solicitar un número entero al usuario, si el usuario proporciona un número mayor a 20, multiplicar 5 por 30 y mostrar el resultado, si es menor o igual a 20 pero mayor a 0, sumar 5 más 10 y sumar el resultado y si es menor o igual a 0, volver a solicitar el valor.**

Datos de entrada: Número proporcionado por el usuario.

Datos de salida: La suma de 5 más 10 si el número es mayor a 0, pero menor a 20 y 150 si es mayor a 20

Restricciones: El número debe ser un número naturales

Algoritmo:

1-Inicio

2-Solicitar el número, ir al paso 3.

3-Verificar el número dado, si es decimal, o menor a 0, ir a paso 2, de no serlo, ir a paso 4.

4-verificar que, si el número dado es mayor a 20, de serlo, ir a paso 6, caso contrario, ir a paso 5.

5-Verificar que el número es menor a 20 y mayor a 0, ir a paso 7.

6-Multiplicar 5 por 30 y mostrar el resultado, ir a paso 8.

7-Sumar 5 más 10 y mostrar el resultado, ir a paso 8

8-Fin

|  |  |  |
| --- | --- | --- |
| Iteración | Dato de entrada | Salida |
| 1 | -2.3 | - |
| 2 | 6 | 15 |
| 3 | 34 | 150 |

**Ejercicio 1:**

PROBLEMA: Seguir el algoritmo para obtener una figura

ENTRADA: Hoja tamaño carta en limpio, regla y lápiz.

SALIDA: Figura correcta.

Algoritmo

1. Dibuja una V invertida. Empieza desde el lado izquierdo, sube, y baja hacia el

lado derecho, no levantes el lápiz.

2. Ahora dibuja una línea en ángulo ascendente hacia la izquierda. Debe cruzar la

primera línea más o menos a 1/3 de la altura. Todavía no levantes el lápiz del

papel.

3. Ahora, dibuja una línea horizontal hacia la derecha. Debe cruzar la V invertida

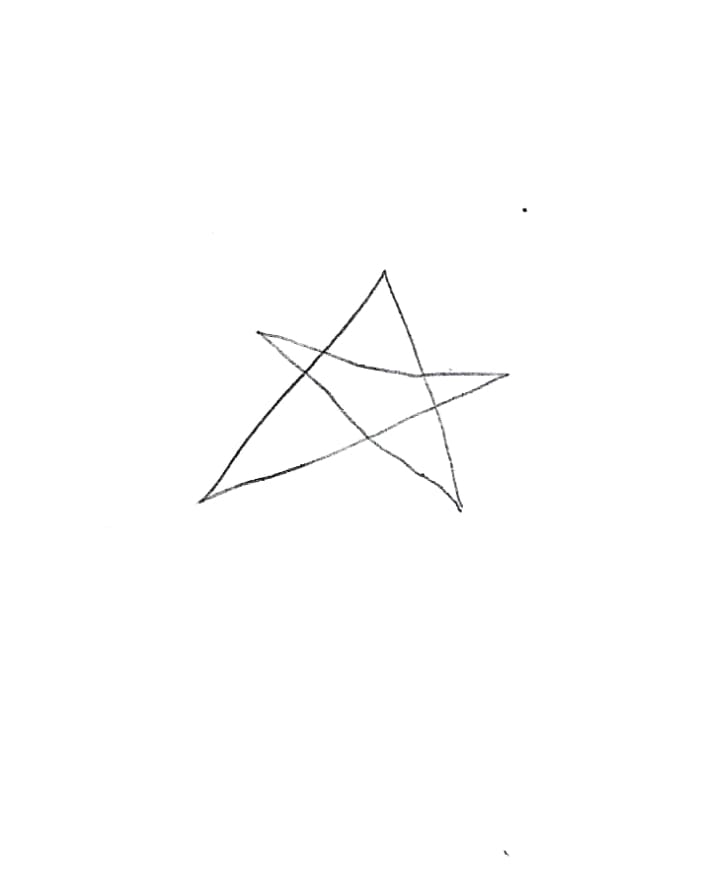
más o menos a 2/3 de la altura total. Sigue sin levantar el lápiz.

4. Dibuja una línea en un ángulo descendente hasta el punto de inicio. Las líneas

deben unirse.

5. Ahora ya puedes levantar el lápiz del papel. Has terminado la estrella de 5

puntas.



El algoritmo, aunque resulto confuso, da el resultado esperado

**Ejercicio 2:**

PROBLEMA: Seguir el algoritmo para obtener una figura

ENTRADA: Hoja tamaño carta en limpio, regla y lápiz.

SALIDA: Figura correcta.

Algoritmo

1. Empieza dibujando un círculo con un compás. Coloca un lápiz en el compás.

Coloca la punta del compás en el centro de una hoja de papel.

2. Ahora gira el compás, mientras mantienes la punta apoyada en el papel. El lápiz

dibujará un círculo perfecto alrededor de la punta del compás.

3. Marca un punto en la parte superior del círculo con el lápiz. Ahora, coloca la

punta del compás en la marca. No cambies el radio del compás con que hiciste el

círculo.

4. Gira el compás para hacer una marca en el propio círculo hacia la izquierda. Haz

una marca también en el lado derecho.

5. Ahora, coloca la punta del compás en uno de los puntos. Recuerda no cambiar el

radio del compás. Haz otra marca en el círculo.

6. Continúa moviendo la punta del compás a las otras marcas, y continúa hasta que

tengas 6 marcas a la misma distancia unas de otras. Ahora, ya puedes dejar tu

compás a un lado.

7. Usa una regla para crear un triángulo que empiece en la marca superior del

círculo. Coloca el lápiz en la marca superior. Ahora dibuja una línea hasta la

segunda marca por la izquierda. Dibuja otra línea, ahora hacia la derecha,

saltándote la marca de la parte más baja. Complementa el triángulo con una línea

hacia la marca superior. Así completarás el triángulo.

8. Crea un segundo triángulo empezando en la marca en la base del círculo. Coloca

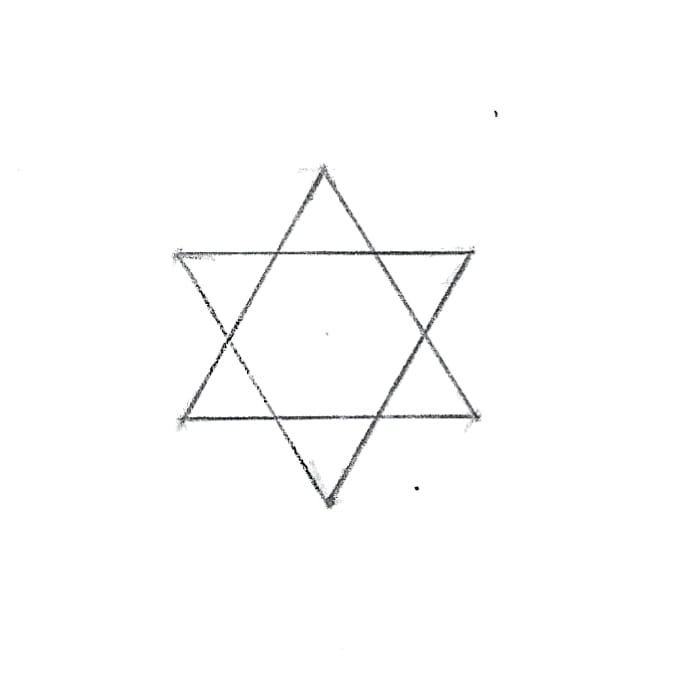
el lápiz en la marca inferior. Ahora conéctala con la segunda marca hacia la

izquierda. Dibuja una línea recta hacia la derecha, saltándote el punto superior.

Completa el segundo triángulo dibujando una línea hasta la marca en la parte

inferior.

9. Borra el círculo. Has terminado de dibujar tu estrella de 6 puntos.



El algoritmo resulta fácil de entender y es correcto

**Conclusiones:**

El algoritmo representa la parte fundamental al elaborar un programa, pues maraca los pasos a seguir para llegar a las soluciones deseadas, para ello se requiere el total entendimiento del problema, además de seguir las pautas para crear un buen algoritmo, además de evitar caer en una ambigüedad, cualquiera que esta sea.

Personalmente, me resultó fácil llevar esta práctica a cabo, puesto que ya tenía cierta experiencia en este tema en específico.

**Bibliografía:**

García E. et al. (2022) *Manual de Prácticas del laboratorio de Fundamento de programación*, recuperado de http://lcp02.fi-b.unam.mx/